DW03_lfsr_scnto

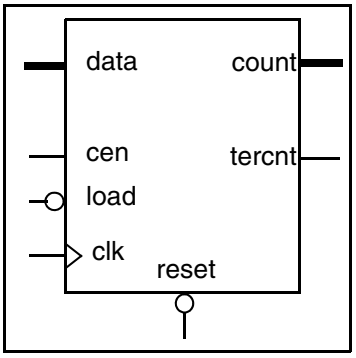
LFSR Counter with Static Count-to Flag

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

Revision History

- Parameterized count-to value to indicate when the counter reaches a specified value
- Parameterized word length
- High speed, area-efficient
- Asynchronous reset
- Terminal count flag



Description

DW03_lfsr_scnto is a parameterized word-length up counter with a static count-to flag. DW03_lfsr_scnto implements a counter as LFSR (linear feedback shift register) which also acts as a pseudo-random counter constructed as primitive characteristic polynomials.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
data	*width* bits	Input	Input data
load	1 bit	Input	Input load, active low
cen	1 bit	Input	Input count enable
clk	1 bit	Input	Clock
reset	1	Input	Asynchronous reset, active low
count	*width* bits	Output	Output count bus
tercnt	1 bit	Output	Output terminal count

Table 1-2 Parameter Description

Parameter	Values[a]	Function
width	2 to 50	Word length of counter
count_to	1 to $2^{width - 2}$	count_to bus

a. The upper bound of the legal range is a guideline to ensure reasonable compile times.

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Synthesis model	DesignWare

Table 1-4 Simulation Models

Model	Function
DW03.DW03_LFSR_SCNTO_CFG_SIM	Design unit name for VHDL simulation
dw/dw03/src/DW03_lfsr_scnto_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW03_lfsr_scnto.v	Verilog simulation model source code

Table 1-5 Counter Operation Truth Table

reset	load	cen	Operation
0	X	X	Reset
1	0	1	Load
1	X	0	Standby
1	1	1	Count

The shift register is fed back from two or more taps. The number of taps does not increase with a large counter *width*.

A LFSR counter runs faster than a binary counter in synchronous systems because the first bit is calculated and the remaining bits are shifted.

DW03_lfsr_scnto can be used in built-in test circuitry in VLSI chips and used as a modulus counter; see Figure 1-1 on page 3.

Counter Function

The `width` is a generic parameter with an integer value ranging from 1 to 50.

The counter is loaded with `data` by asserting `load` (low) and applying data to `data`. The data load operation is synchronous with respect to the positive edge of `clk`.

The `count_to` is an integer parameter of pseudo-random binary sequences that ranges from 1 to $2^{width - 2}$. For a list of the primitive polynomials, see the Logic–Sequential Overview.

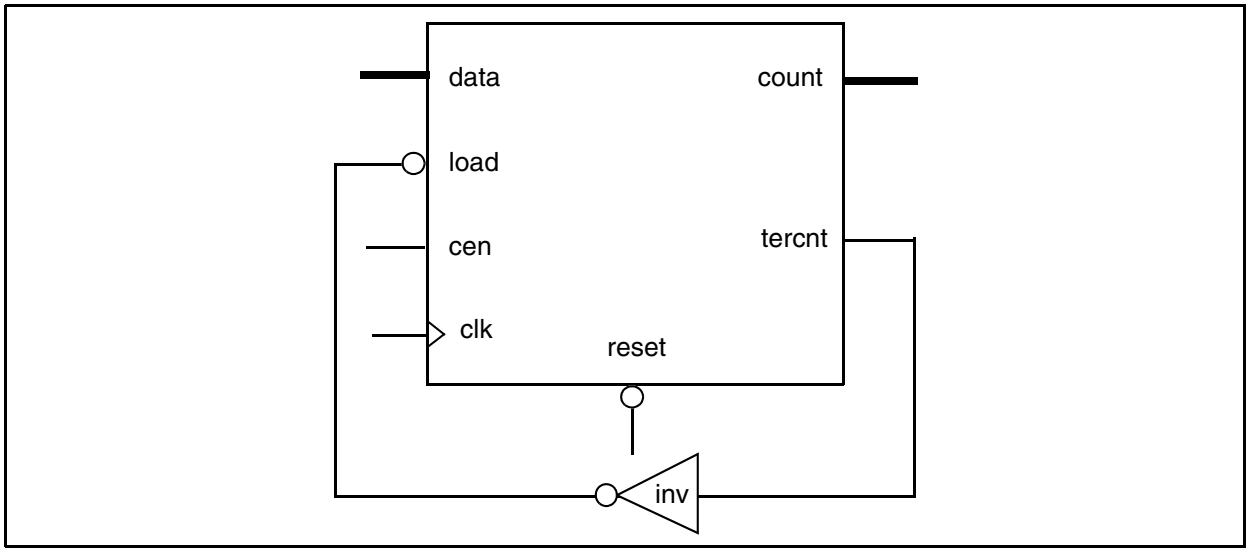
The count enable pin, `cen`, is active high. When `cen` is high, the counter is active. When `cen` is low, the counter is disabled and `count` remains at the same value.

The `reset` signal is an asynchronous reset that is active low. When `reset` is low, the counter output is "00...00". When `reset` is high, the counter operates normally.

The `count` is the output port of pseudo-random binary sequences, ranging from *width* - 1 to 0. A value of $2^{width - 2}$ (“11...11”) is an illegal state; therefore, the counter stops at “11...11”.

The `tercnt` is an output terminal count signal, active high. The `tercnt` output goes high for one clock cycle to indicate the different number of clock cycles between starting count to `count_to` value.

Figure 1-1 Counter Application: “count_to”



Timing Diagrams

Figure 1-2 and Figure 1-3 show various timing conditions for DW03_lfsr_scnto.

Figure 1-2 Functional Operation: load and count_enable Sequence

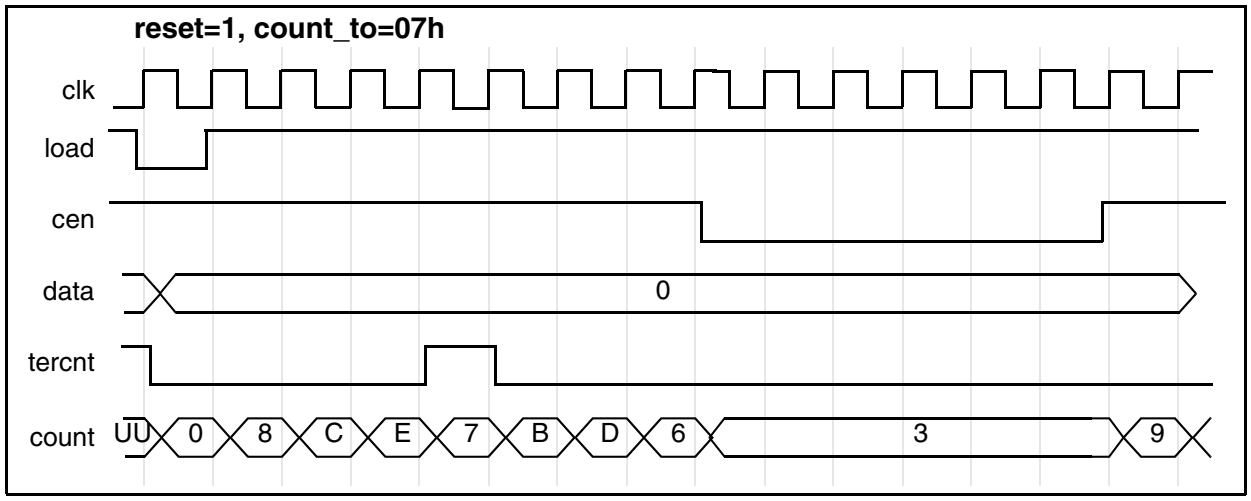
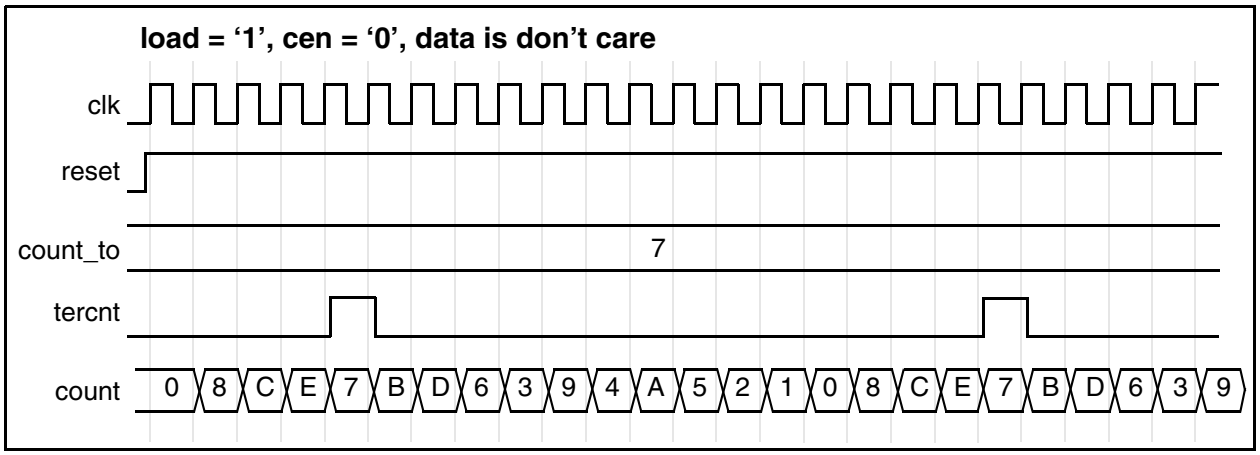


Figure 1-3 Reset Sequence



Related Topics

- Logic – Sequential Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_foundation_comp.all;

entity DW03_lfsr_scnto_inst is
  generic ( inst_width    : INTEGER := 8;
            inst_count_to : INTEGER := 8);
  port ( inst_data   : in std_logic_vector(inst_width-1 downto 0);
         inst_load   : in std_logic;
         inst_cen    : in std_logic;
         inst_clk    : in std_logic;
         inst_reset  : in std_logic;
         count_inst  : out std_logic_vector(inst_width-1 downto 0);
         tercnt_inst : out std_logic );
end DW03_lfsr_scnto_inst;

architecture inst of DW03_lfsr_scnto_inst is
begin

  -- Instance of DW03_lfsr_scnto
  U1 : DW03_lfsr_scnto
    generic map ( width => inst_width, count_to => inst_count_to )
    port map ( data => inst_data, load => inst_load,
               cen => inst_cen, clk => inst_clk, reset => inst_reset,
               count => count_inst, tercnt => tercnt_inst );
end inst;

-- pragma translate_off
configuration DW03_lfsr_scnto_inst_cfg_inst of DW03_lfsr_scnto_inst is
  for inst
  end for; -- inst
end DW03_lfsr_scnto_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW03_lfsr_scnto_inst( inst_data, inst_load, inst_cen, inst_clk,
                             inst_reset, count_inst, tercnt_inst );

  parameter width = 8;
  parameter count_to = 8;

  input [width-1 : 0] inst_data;
  input inst_load;
  input inst_cen;
  input inst_clk;
  input inst_reset;
  output [width-1 : 0] count_inst;
  output tercnt_inst;

  // Instance of DW03_lfsr_scnto
  DW03_lfsr_scnto #(width, count_to)
    U1 ( .data(inst_data), .load(inst_load), .cen(inst_cen), .clk(inst_clk),
         .reset(inst_reset), .count(count_inst), .tercnt(tercnt_inst) );

endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
March 2019	DWBB_201903.0	■ Removed minPower designation from this datasheet
January 2019	DWBB_201806.5	■ Updated example in "HDL Usage Through Component Instantiation - VHDL" on page 5 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com